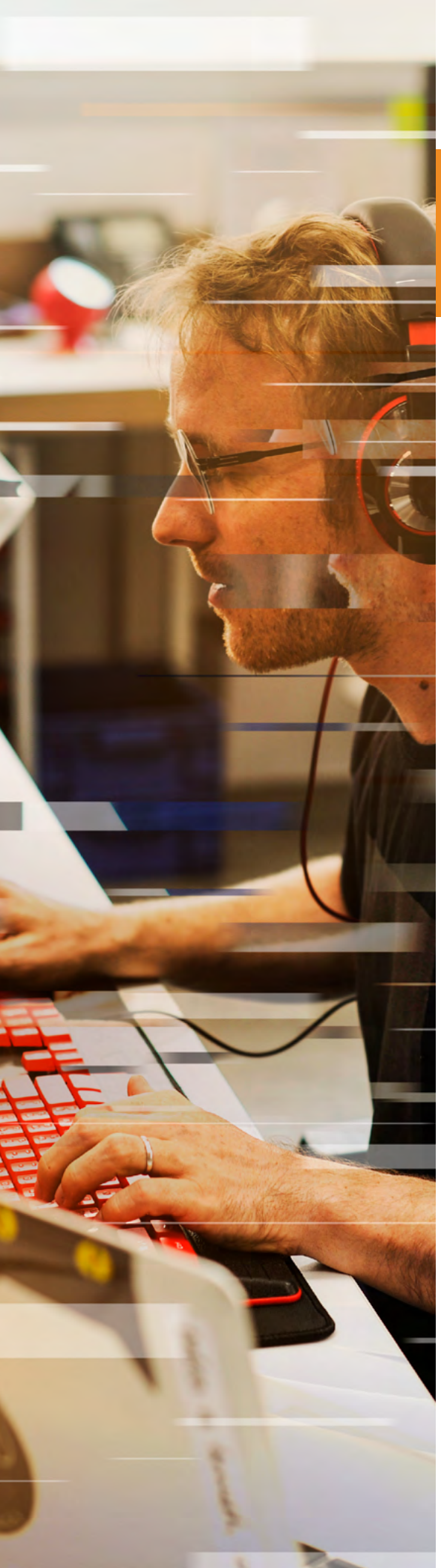
The recipe for hyperfast DevOps instrumentation

An e-guide to infrastructure as code



Why take infrastructure out of the physical world?

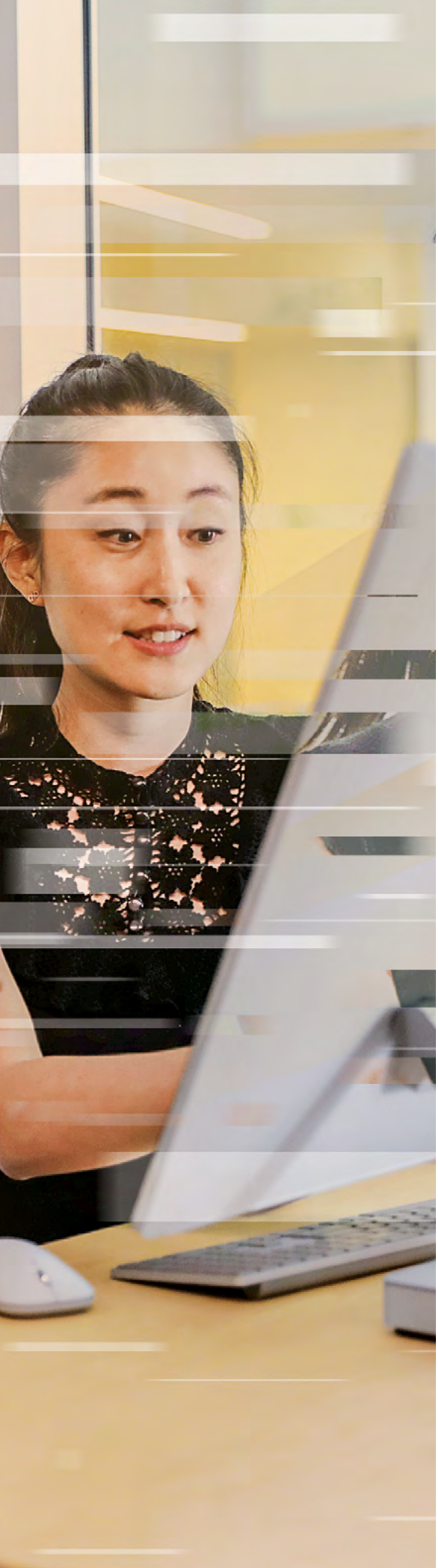
Up to now, setting up instrumentation for new projects was a time-consuming process that required resources from development and operations teams. Systems administrators would create artifacts by hand, logging in and changing config files for each needed environment.

They might start with scripts to get basic automation in place and to do repetitive tasks. Then, after troubleshooting, they would hand the single space off to development to work on.

What if you could take that to the next level and stand up a machine, group of machines, or an entire environment simply using the codes, or the DNA, of your infrastructure? The idea is that you establish the basic structure that works and then allow the system to replicate it automatically.

That is infrastructure as code (IaC).

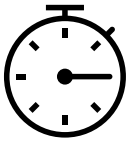
The process for IaC is simple. To get started, you can use prewritten code, or "recipes" and cookbooks that other people have already developed. You can also write your own, describing your own infrastructure using a domain-specific language (DSL), a powerful and easy-to-learn platform based on Ruby.



And once you're expressing your infrastructure as code, you can repeatedly, consistently, and reliably stand up any number of virtual machines, cloud instances, or containers, and do each in a matter of minutes rather than hours—or days.

This not only speeds immediate work, but also can greatly accelerate migration to the cloud as part of a DevOps transition. For example, if you're trying to be more agile by moving from hardware on-premises to virtual machines, expressing your infrastructure as code allows you to move fast and inexpensively. Your code is very portable, so you can run it on any platform—and that helps give all your projects a common foundation.

Whether your infrastructure is in the cloud, on-premises, or in a hybrid environment, IaC helps you easily and quickly adapt to your business's changing needs. Whether you have five or 5,000 servers, infrastructure described as code is flexible, versionable, human-readable, and testable. And when you establish a DevOps culture, you get three critical elements:



Speed

Very few organizations still operate in a single server location, cloud, or container technology. That means infrastructure creation and management can be even more of a resource drain.

With IaC, you can do any number of migrations and stand up any number of machines without needing to have different platforms pulled up or hiring new resources. Deploying a single DevOps process helps enable a high-velocity organization and gives you a strong competitive advantage that's hard to achieve otherwise.

Innovation

Like DNA, IaC can be extremely repeatable and reliable. As long as your code foundation is strong, it becomes easy to stand up all the infrastructure that's required to run the application again and again.

Once that's in place, IaC unlocks many other possibilities for automated testing, built-in security and audit checks, and the ability to do things like blue-green deployment. It also enables you to keep your app running continuously, 24/7, without having a maintenance window.

Risk reduction

Increased reliability is a change that stakeholders, customers, and even the general public notice right away. With so many applications and products launching with bugs included, the ability to maintain your infrastructure and increased certainty that the app will work all the time can be a strong differentiator externally. And internally, when you're reducing your infrastructure build times by as much as 90 percent and reducing risk, you're having real impact.

Case study: DevOps culture at flight speed

"Writing code absent context is meaningless," says one CIO. "The context is all about time. When the developer enters that line of code, he or she should know the impact that's going to have on operations, on customer experience, on business results. It's all in the name of reducing time—time to develop, time to market, even time to board."

> "Writing code absent context is meaningless."
>
> — One CIO

Airline culture is about people and speed. So adopting a DevOps culture was a natural choice. This leading airline has made a large investment in technology to create a smooth customer experience, from trip planning in a five-star app that integrates with digital luggage tags to the future of kiosk-free check-in. And it's all driven by time.

The CIO sums it up this way: "I don't care whether you call it DevOps. When it's all said and done, what we're talking about is getting heterogeneous, collocated teams—putting them together—to deliver outcomes at speed."

That's what transforms an airline—and any organization.

The keys to getting it right

Start small

Pick a use case that shows the value of IaC—something with an easily identifiable and manageable scope.

Learn what's possible

One of the best ways to evaluate a new solution and get some training is to play with it. Then, once you're comfortable, automate a single use case that used to take you minutes or hours or days to do. You'll see the value right away.

Spread the word

IaC and DevOps are about more than just tooling. They're about changing culture. It's easier to bring attention to the issues facing your organization if you have a solution you're familiar with. Now that you can quantify the benefits and show the results, you can get more teams and leadership behind the change.

NCR: Go fast. Go big. A Super Bowl story.

Global tech leader NCR needed to support a massive Super Bowl promotion for a customer, a task that would have required the company to build a new product environment with many times more servers than had ever been there before—and to do it in short order. In addition, NCR

needed to provision a huge number of servers very rapidly and make them consistent with servers previously in production, as well as able to reliably handle a colossal one-time surge in activity. All without risk. With IaC, that was all possible.

NCR did it, and well within deadline. Using IaC, the team was able to attain a level of speed and innovation never dreamed of a couple of years ago. With this initiative, NCR was able to move into a new era of dramatically faster deployments as part of its DevOps transition.

The security question—answered

Compliance and security are very real, everyday problems for your customers. Companies require speed, velocity, and safety to compete in the digital marketplace. There's immense pressure to move faster to deliver value for IT. When it comes to information security (InfoSec), there's a big gap between perception and reality. According to Gartner, 88 percent of IT professionals believe InfoSec policies inhibit agility and speed, and 77 percent of InfoSec professionals agree that their policies inhibit agility and speed. The reality is that the exploits cause dramatic downtime: 50 percent of teams need multiple days or more to remediate them, and 28 percent of teams need weeks or months. Ninety-nine percent of vulnerabilities exploited will be known to InfoSec for one year or more, and since 2014, more than 88 percent of exploits observed use only nine known vulnerabilities.

By turning compliance into code, Chef helps customers build in compliance automation for systems and applications and deliver IT continuously at scale to the Microsoft Azure cloud. Compliance built into automation becomes repeatable, reliable, and verifiable. Chef InSpec enables customers to clearly express statements of policy, move risk to build/test from runtime, find issues early, write code quickly, and run code anywhere.

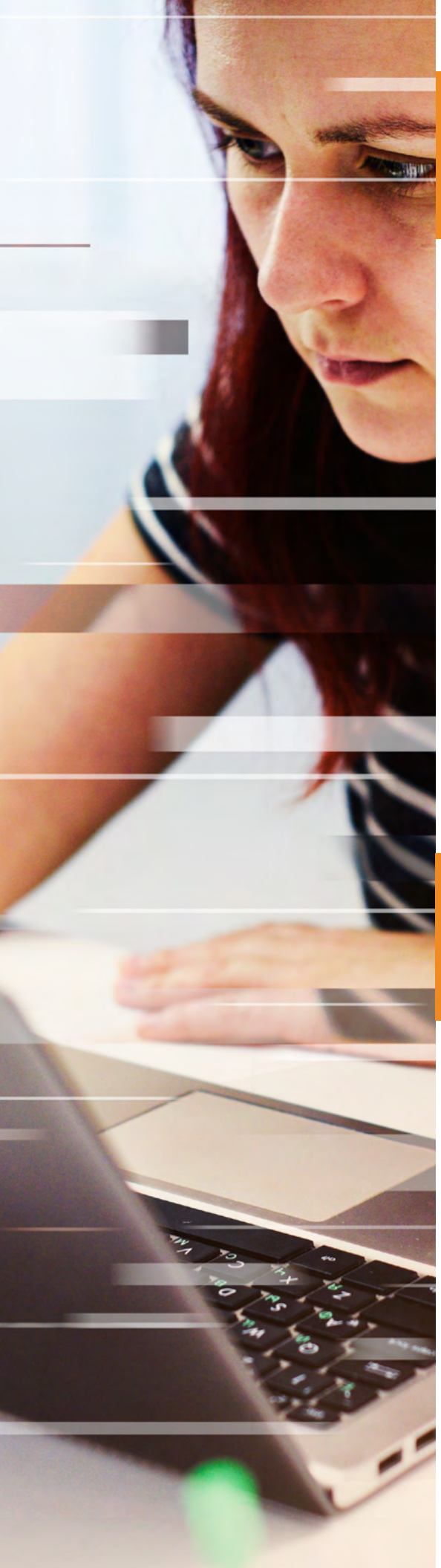
Three IaC pitfalls to avoid

1. Unbridled enthusiasm. The dangers of speed.
When you can create an entire virtual machine in virtually no time, and make it work for any number of instances very quickly after that, the temptation is to try to change the entire organization's virtual existence right away. Recommendation: start with a smaller, well-defined use case, solve it, and then build from there.

2. The inertia factor
Once you have established how IaC can bring real improvements to your business, you will encounter the opposite of unbridled enthusiasm in the form of risk-averse behavior or "We've always done it this way."

The beauty of having an established case before socializing IaC is that you can make specific statements like "a savings of $100,000 per quarter" and "an improvement in speed of 350 percent." Not to mention the fact that velocity equals competitive advantage, and if you're not dramatically improving speed to market, others are.

3. The Lamborghini syndrome
One final tip. Using IaC, you'll quickly find that it's so simple to spin up to 10 virtual machines on a whim. That's great. But remember, each machine is using compute cycles that cost money every hour. So, though it may be great fun to drive, like a Lamborghini, all that power can get expensive fast. So IaC responsibly.



Change the tools. Change the culture.

Many organizations are very siloed. There's usually a systems administrator or operations team, and then there are the devs. The two groups don't always speak the same language. Sometimes they even feel like they're at odds. The developers are more used to testing their software using tools. The sysadmins are more concerned with day-to-day operations like "Is the app up or down? Are there any alarms that I have to deal with?" IaC can help bring those two worlds together. It applies agile and lean principles to the world of infrastructure for greater collaboration.

Now, using DevOps and IaC, different groups can work together toward common business goals. Both dev and ops feel a strong stake in the outcomes, and they're empowered to communicate with each other. Once everyone is accustomed to working in that culture, they're no longer merely meeting each other's SLAs and escalating when things go wrong. They're solving real problems and removing roadblocks for better systems.

Chef and Microsoft Azure enable the DevOps transition

Chef partners with Microsoft Azure for powerful solutions that automatically transform infrastructure into code to increase velocity while reducing risk through a suite of automation tools that give you workflow automation for continuous deployment, automated testing for compliance, and security. Together, they provide the platform and the computing power to continuously automate infrastructure, applications, and compliance across your environment, no matter its size.

CHEF Microsoft Azure

Whether you're operating in the cloud, on-premises, or in a hybrid environment, Chef and Azure are powerful allies in your shift to DevOps. Chef Automate works with Microsoft Operations Management Services (OMS) so that customers can get a single view of all their alerts in Azure Log Analytics. Furthermore, if you are considering Azure Stack as your hybrid cloud solution, Chef Automate is available for use on Azure Stack.

Take the next step in automating your DevOps journey.

Visit Chef.io

Get the full DevOps picture

Take advantage of an array of DevOps resources.

Find out how DevOps adoption can improve velocity, quality, and organizational culture at every phase of development.
Explore more

CHEF Microsoft Azure